



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/688,640	10/17/2003	Antti Kokkinen	14322US02	1958
23446 7590 08/21/2007 MCANDREWS HELD & MALLOY, LTD 500 WEST MADISON STREET SUITE 3400 CHICAGO, IL 60661			EXAMINER WANG, BEN C	
			ART UNIT 2192	PAPER NUMBER
			MAIL DATE 08/21/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/688,640	Applicant(s) KOKKINEN, ANTTI	
	Examiner Ben C. Wang	Art Unit 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 23 May 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-24 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-24 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Applicant's amendment dated May 23, 2007, responding to the Office action mailed Feb. 23, 2007 provided in the rejection of claims 1-24, wherein claims 2-24 are remained as original, claim 1 is canceled.

Claims 1-24 remain pending in the application and which have been fully considered by the examiner.

Applicant's arguments with respect to claims rejection have been fully considered but are moot in view of the new grounds of rejection – see both *Gu et al.*, *Liu et al.*, and *Meyerson et al.*, arts made of record, as applied hereto.

Claim Rejections – 35 USC § 102(e)

The following is quotation of 35 U.S.C. 102(e) which form the basis for all obviousness rejections set forth in this office action:

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. Claims 1-9, 12, and 19-24 are rejected under 35 U.S.C. 102(e) as being anticipated by Gu et al. (Pat. No. US 6,925,467 B2) (hereinafter 'Gu' - art made of record)

3. **As to claim 1**, Gu discloses a method for updating software in an electronic device (e.g., Col. 1, Lines 33-37 – in mobile wireless devices, a real-time operating

system (RTOS) is typically used in which all software components are linked as a single large file; further, no file system support is typically provided in these mobile wireless devices; Abstract, Lines 1-9 – a method for performing differencing and updating between electronic files is provided; a byte-level file differencing algorithm receives two byte streams corresponding to an original file and a new file. The new file includes updated and revised versions of the original file. The file differencing algorithm determines a longest common sub-string (LCS) between the two byte streams and divides each of the two byte streams into sub-streams. The sub-streams includes the LCS along with prefix and suffix sub-streams to the LCS), the method comprising: generating an update package for updating at least one software application (Fig. 1, element 116 of “Delta File”) being generated based upon at least one reference software installed on the electronic device (e.g., Fig. 1 – showing file differencing and updating provided by byte-level file differencing and updating (FDU) algorithms, element of “Hosted Original File”); and updating the at least one software application using the update package and the reference software; and wherein the updating leaves the at least one reference software unchanged (e.g., Col. 4, Lines 5-17 – the delta file [the update package] is transferred or transmitted to another computer system via the communication path. Prior to transfer, the delta file may be compressed using compression techniques know in the art, but is not so limited. The file updating algorithm hosted on the receiving computer system use the delta file along with the hosted original file [one reference software] to generate or create a copy of the new file.

This copy of new file is then used to update the original file hosted on the client device that is targeted for revision or updating).

4. **As to claim 12**, Gu discloses a system for updating software, the system comprising: an electronic device capable of having software installed thereon (e.g., Abstract, Lines 1-2 – a method for performing differencing and updating between electronic files is provided); a software delivery device for receiving and installing a reference software to the electronic device if the electronic device does not have the reference software previously installed (e.g., Col. 1, Lines 33-37 – in mobile wireless devices, a real-time operating system (RTOS) is typically used in which all software components are linked as a single large file; further, no file system support is typically provided in these mobile wireless devices; in addition, the single large file needs to be preloaded, or embedded, into the device using a slow communication link like a radio, infrared, or serial link); and the software delivery device receiving and delivering at least one update package to the electronic device, wherein the reference software facilitates, using the at least one update package, at least one update to application software installed on the electronic device (e.g., Col. 4, Lines 5-17 – the delta file [the update package] is transferred or transmitted to another computer system via the communication path. Prior to transfer, the delta file may be compressed using compression techniques known in the art, but is not so limited. The file updating algorithm hosted on the receiving computer system use the delta file along with the hosted original file [one reference software] to generate or create a copy of the new file.

This copy of new file is then used to update the original file hosted on the client device that is targeted for revision or updating).

5. **As to claim 21**, Gu discloses a method for updating software in an electronic device the method comprising: generating a first update package for updating at least one software application, the first update package being generated based upon difference information between first and second software versions (e.g., Col. 4, Lines 5-17 – the delta file [the update package] is transferred or transmitted to another computer system via the communication path. Prior to transfer, the delta file may be compressed using compression techniques know in the art, but is not so limited. The file updating algorithm hosted on the receiving computer system use the delta file along with the hosted original file [one reference software] to generate or create a copy of the new file. This copy of new file is then used to update the original file hosted on the client device that is targeted for revision or updating); generating a second update package for updating the at least one software application, the second update package being generated based upon difference information between first and third software versions (e.g., Col. 4, Lines 5-17 – the delta file [the update package] is transferred or transmitted to another computer system via the communication path. Prior to transfer, the delta file may be compressed using compression techniques know in the art, but is not so limited. The file updating algorithm hosted on the receiving computer system use the delta file along with the hosted original file [one reference software] to generate or create a copy of the new file. This copy of new file is then used to update the original file hosted on the client device that is targeted for revision or updating); generating a third

Art Unit: 2192

update package for updating the at least one software application, the third update package being generated based upon difference information between the first and second update packages; and updating the at least one software application using the third update package (e.g., Col. 3, Lines 40-51 – the new file 112 is generally an updated or revised version of the original file 110, but is not so limited; the electronic files 110 and 112 include software files including dynamic link library files, shared object files, embedded software components, firmware files, executable files, data files including hex data files, system configuration files, and files including personal use data, but are not so limited; Since any type of file can be regarded as byte stream, hereafter a file can be described as a byte stream, depending the content).

6. **As to claim 22**, Gu discloses a method for updating software in an electronic device, the method comprising: generating a first update package for updating at least one software application, the first update package being generated based upon difference information between a first software version and a reference software corresponding to the at least one software application (e.g., Col. 4, Lines 5-17 – the delta file [the update package] is transferred or transmitted to another computer system via the communication path. Prior to transfer, the delta file may be compressed using compression techniques know in the art, but is not so limited. The file updating algorithm hosted on the receiving computer system use the delta file along with the hosted original file [one reference software] to generate or create a copy of the new file. This copy of new file is then used to update the original file hosted on the client device

Art Unit: 2192

that is targeted for revision or updating); generating a second update package for updating the at least one software application, the second update package being generated based upon difference information a second software version and the reference software corresponding to the at least one software application (e.g., Col. 4, Lines 5-17 – the delta file [the update package] is transferred or transmitted to another computer system via the communication path. Prior to transfer, the delta file may be compressed using compression techniques know in the art, but is not so limited. The file updating algorithm hosted on the receiving computer system use the delta file along with the hosted original file [one reference software] to generate or create a copy of the new file. This copy of new file is then used to update the original file hosted on the client device that is targeted for revision or updating); generating a third update package for updating the at least one software application, the third update package being generated based upon difference information between the first and second update packages; and updating the at least one software application using the third update package (e.g., Col. 3, Lines 40-51 – the new file 112 is generally an updated or revised version of the original file 110, but is not so limited; the electronic files 110 and 112 include software files including dynamic link library files, shared object files, embedded software components, firmware files, executable files, data files including hex data files, system configuration files, and files including personal use data, but are not so limited; Since any type of file can be regarded as byte stream, hereafter a file can be described as a byte stream, depending the content).

7. **As to claim 23**, Gu discloses a system for updating software, the system comprising: an electronic device capable of having software installed thereon; a first update package generator for generating update packages based upon difference information between different versions of software (e.g., Col. 4, Lines 5-17 – the delta file [the update package] is transferred or transmitted to another computer system via the communication path. Prior to transfer, the delta file may be compressed using compression techniques known in the art, but is not so limited. The file updating algorithm hosted on the receiving computer system use the delta file along with the hosted original file [one reference software] to generate or create a copy of the new file. This copy of new file is then used to update the original file hosted on the client device that is targeted for revision or updating); a second update package generator for generating update packages based upon difference information between different update packages; and a software delivery device for delivering at least one update package generated based upon difference information between different update packages to the electronic device (e.g., Col. 3, Lines 40-51 – the new file 112 is generally an updated or revised version of the original file 110, but is not so limited; the electronic files 110 and 112 include software files including dynamic link library files, shared object files, embedded software components, firmware files, executable files, data files including hex data files, system configuration files, and files including personal use data, but are not so limited; Since any type of file can be regarded as byte stream, hereafter a file can be described as a byte stream, depending the content).

Art Unit: 2192

8. **As to claim 24**, Gu discloses a system for updating software, the system comprising: an electronic device capable of having software installed thereon; a first update package generator for generating update packages based upon difference information between a version of software and a reference software corresponding to at least one software application (e.g., Col. 4, Lines 5-17 – the delta file [the update package] is transferred or transmitted to another computer system via the communication path. Prior to transfer, the delta file may be compressed using compression techniques known in the art, but is not so limited. The file updating algorithm hosted on the receiving computer system use the delta file along with the hosted original file [one reference software] to generate or create a copy of the new file. This copy of new file is then used to update the original file hosted on the client device that is targeted for revision or updating); a second update package generator for generating update packages based upon difference information between different update packages; and a software delivery device for delivering at least one update package generated based upon difference information between different update packages to the electronic device (e.g., Col. 3, Lines 40-51 – the new file 112 is generally an updated or revised version of the original file 110, but is not so limited; the electronic files 110 and 112 include software files including dynamic link library files, shared object files, embedded software components, firmware files, executable files, data files including hex data files, system configuration files, and files including personal use data, but are not so limited; Since any type of file can be regarded as byte stream, hereafter a file can be described as a byte stream, depending the content).

9. **As to claim 2**, Gu discloses the method wherein generating an update package for updating the at least one software application based upon the at least one reference software installed on the electronic device comprises: accessing a copy of the at least one reference software; retrieving a copy of the at least one software application; and generating an update package (e.g., Col. 4, Lines 5-17 – the delta file [the update package] is transferred or transmitted to another computer system via the communication path. Prior to transfer, the delta file may be compressed using compression techniques known in the art, but is not so limited. The file updating algorithm hosted on the receiving computer system use the delta file along with the hosted original file [one reference software] to generate or create a copy of the new file. This copy of new file is then used to update the original file hosted on the client device that is targeted for revision or updating).

10. **As to claim 3**, Gu discloses the method wherein generating an update package for updating at least one software application based upon the at least one reference software installed on the electronic device comprises: accessing a copy of the at least one reference software; retrieving a copy of each of multiple versions of the at least one software application and generating an update package comprising all transitions between the retrieved versions of the at least one software application (e.g., Col. 4, Lines 5-17 – the delta file [the update package] is transferred or transmitted to another computer system via the communication path. Prior to transfer, the delta file may be

Art Unit: 2192

compressed using compression techniques known in the art, but is not so limited. The file updating algorithm hosted on the receiving computer system uses the delta file along with the hosted original file [one reference software] to generate or create a copy of the new file. This copy of new file is then used to update the original file hosted on the client device that is targeted for revision or updating).

11. **As to claim 4**, Gu discloses the method further comprising updating multiple update versions of the at least one software application installed on the electronic device is performed using a single update package (e.g., Col. 1, Lines 33-37 – in mobile wireless devices, a real-time operating system (RTOS) is typically used in which all software components are linked as a single large file; further, no file system support is typically provided in these mobile wireless devices; Abstract, Lines 1-9 – a method for performing differencing and updating between electronic files is provided; a byte-level file differencing algorithm receives two byte streams corresponding to an original file and a new file. The new file includes updated and revised versions of the original file. The file differencing algorithm determines a longest common sub-string (LCS) between the two byte streams and divides each of the two byte streams into sub-streams. The sub-streams includes the LCS along with prefix and suffix sub-streams to the LCS).

12. **As to claim 5**, Gu discloses the method further comprising installing the at least one software application and the at least one reference software as part of a single installation (e.g., Col. 1, Lines 33-37 – in mobile wireless devices, a real-time operating

Art Unit: 2192

system (RTOS) is typically used in which all software components are linked as a single large file; further, no file system support is typically provided in these mobile wireless devices; Abstract, Lines 1-9 – a method for performing differencing and updating between electronic files is provided; a byte-level file differencing algorithm receives two byte streams corresponding to an original file and a new file. The new file includes updated and revised versions of the original file. The file differencing algorithm determines a longest common sub-string (LCS) between the two byte streams and divides each of the two byte streams into sub-streams. The sub-streams includes the LCS along with prefix and suffix sub-streams to the LCS).

13. **As to claim 6**, Gu discloses the method further comprising updating the at least one reference software and updating the at least one software application as part of a single update (e.g., Col. 1, Lines 33-37 – in mobile wireless devices, a real-time operating system (RTOS) is typically used in which all software components are linked as a single large file; further, no file system support is typically provided in these mobile wireless devices; Abstract, Lines 1-9 – a method for performing differencing and updating between electronic files is provided; a byte-level file differencing algorithm receives two byte streams corresponding to an original file and a new file. The new file includes updated and revised versions of the original file. The file differencing algorithm determines a longest common sub-string (LCS) between the two byte streams and divides each of the two byte streams into sub-streams. The sub-streams includes the LCS along with prefix and suffix sub-streams to the LCS).

14. **As to claim 7**, Gu discloses the method wherein the at least one software application comprises a plurality of software applications, and the at least one reference software comprises a plurality of reference software (e.g., Col. 3, Lines 40-51 – the new file 112 is generally an updated or revised version of the original file 110, but is not so limited; the electronic files 110 and 112 include software files including dynamic link library files, shared object files, embedded software components, firmware files, executable files, data files including hex data files, system configuration files, and files including personal use data, but are not so limited; Since any type of file can be regarded as byte stream, hereafter a file can be described as a byte stream, depending the content).

15. **As to claim 8**, Gu discloses the method further comprising: identifying a software application needing updating from the plurality of software applications installed on the electronic device; identifying whether a reference software corresponding to the software application needing updating is present on the electronic device, wherein if the reference software is not present, then installing the software application and an associated reference software in a single update on the electronic device (e.g., Col. 1, Lines 33-37 – in mobile wireless devices, a real-time operating system (RTOS) is typically used in which all software components are linked as a single large file; further, no file system support is typically provided in these mobile wireless devices; Abstract, Lines 1-9 – a method for performing differencing and updating between electronic files

is provided; a byte-level file differencing algorithm receives two byte streams corresponding to an original file and a new file. The new file includes updated and revised versions of the original file. The file differencing algorithm determines a longest common sub-string (LCS) between the two byte streams and divides each of the two byte streams into sub-streams. The sub-streams includes the LCS along with prefix and suffix sub-streams to the LCS).

16: **As to claim 9**, Gu discloses the method further comprising: identifying a software application needing updating from the plurality of software applications installed on the electronic device; and identifying whether a reference software corresponding to the software application needing updating is present on the electronic device, wherein if the reference software is present, then retrieving an update package for the software application needing updating; verifying the update package; and installing the update package on the electronic device (e.g., Col. 4, Lines 5-17 – the delta file [the update package] is transferred or transmitted to another computer system via the communication path. Prior to transfer, the delta file may be compressed using compression techniques know in the art, but is not so limited. The file updating algorithm hosted on the receiving computer system use the delta file along with the hosted original file [one reference software] to generate or create a copy of the new file. This copy of new file is then used to update the original file hosted on the client device that is targeted for revision or updating).

Art Unit: 2192

17. **As to claim 19**, Gu discloses the system wherein the software delivery device is one of a server (e.g., Col. 22, Lines 27-29) , a CDROM (e.g., Col. 1, Line 31), and a network (e.g., Col. 3, Lines 30-31).

18. **As to claim 20**, Gu discloses the system wherein the electronic device is one of a computer, a digital phone, and a digital camera (e.g., Col. 20, Lines 12-17).

Claim Rejections – 35 USC § 103(a)

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

19. Claims 10-11 and 14-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gu in view of Liu et al., (Pat. No. US 7,143,405 B2) (hereinafter 'Liu' - art made of record)

20. **As to claim 10**, Gu does not explicitly disclose the method further comprising: identifying a software application needing updating from the plurality of software applications installed on the electronic device; determining if the update is needed immediately; and storing the update until the update is needed immediately.

However, in an analogous art of Methods and Arrangements for Managing Devices, Liu discloses the method further comprising: identifying a software application needing updating from the plurality of software applications installed on the electronic device; determining if the update is needed immediately; and storing the update until the update is needed immediately (e.g., Abstract, Lines 5-9 - a method for providing software to a device is provided that includes identifying essential software within the device; essential software including software necessary to operate the device and provide external communications).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Liu into the Gu's system to further provide the method further comprising: identifying a software application needing updating from the plurality of software applications installed on the electronic device; determining if the update is needed immediately; and storing the update until the update is needed immediately in Gu system.

The motivation is that it would further enhance the Gu's system by taking, advancing and/or incorporating Liu's system which offers significant advantages that it would be beneficial to be able to upgrade the down loader code within a managed device; preferably, the methods and arrangements will allow for a failsafe upgrade to all of the code in the managed device as once suggested by Liu (Col. 1, Lines 36-2).

21. **As to claim 11**, Liu discloses the method wherein when the update is determined to be needed immediately, then invoking an update agent to employ at least

Art Unit: 2192

the stored update package and reference software; and updating the software application with the update package (e.g., Abstract, Lines 15-23 – after downloading the new essential software, the method further includes attempting to operate the device and provide external communications using the new essential software, and upon successfully operating the device and providing external communications, using the new essential software to selectively download new non-essential software...).

22. **As to claim 14**, Gu does not disclose the system further comprising an update generating system, the update generating system comprising a loader manager, the loader manager: managing loading of application software and application software version updates from the software delivery device; employing a loader from a loader module; and employing security services to authenticate software being delivered.

However, in an analogous art of Methods and Arrangements for Managing Devices, Liu discloses the system further comprising an update generating system, the update generating system comprising a loader manager, the loader manager: managing loading of application software and application software version updates from the software delivery device; employing a loader from a loader module; and employing security services to authenticate software being delivered (e.g., Col. 2, Line 65 through Col. 3, Line 1; Fig. 1 – an exemplary arrangement having a server device and a (managed) client device; Col. 32-43 – client device may from time-to-time check with server device to determine if any updates to the code/data stored within client device

then client device and server device may immediately begin an updating process, or otherwise selectively establish a time in the future to begin such an updating process).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Liu into the Gu's system to further provide the system further comprising an update generating system, the update generating system comprising a loader manager, the loader manager: managing loading of application software and application software version updates from the software delivery device; employing a loader from a loader module; and employing security services to authenticate software being delivered in Gu system.

The motivation is that it would further enhance the Gu's system by taking, advancing and/or incorporating Liu's system which offers significant advantages that it would be beneficial to be able to upgrade the down loader code within a managed device; preferably, the methods and arrangements will allow for a failsafe upgrade to all of the code in the managed device as once suggested by Liu (Col. 1, Lines 36-2).

23. **As to claim 15**, Liu discloses the system wherein the loader manager further comprises an installation agent for installing application software and downloading files from the software delivery device (e.g., Col. 2, Line 65 through Col. 3, Line 1; Fig. 1 – an exemplary arrangement having a server device and a (managed) client device; Col. 32-43 – client device may from time-to-time check with server device to determine if any updates to the code/data stored within client device then client device and server device

may immediately begin an updating process, or otherwise selectively establish a time in the future to begin such an updating process).

24. **As to claim 16**, Liu discloses the system wherein the loader manager is adapted to: identify an application software needing updating; identify whether reference software associated with the application software needing updating exists; and coordinating an update of the application software and an associated reference software in a single update (e.g., Col. 2, Line 65 through Col. 3, Line 1; Fig. 1 – an exemplary arrangement having a server device and a (managed) client device; Col. 32-43 – client device may from time-to-time check with server device to determine if any updates to the code/data stored within client device then client device and server device may immediately begin an updating process, or otherwise selectively establish a time in the future to begin such an updating process).

25. **As to claim 17**, Liu discloses the system wherein the loader manager is adapted to: retrieve the update package; access contents of the update package; and verify the update package (e.g., Col. 2, Line 65 through Col. 3, Line 1; Fig. 1 – an exemplary arrangement having a server device and a (managed) client device; Col. 32-43 – client device may from time-to-time check with server device to determine if any updates to the code/data stored within client device then client device and server device may immediately begin an updating process, or otherwise selectively establish a time in the future to begin such an updating process).

26. **As to claim 18**, Liu discloses the system wherein the loader manager is adapted to determine immediacy of a needed update for a particular application software (e.g., Col. 2, Line 65 through Col. 3, Line 1; Fig. 1 – an exemplary arrangement having a server device and a (managed) client device; Col. 32-43 – client device may from time-to-time check with server device to determine if any updates to the code/data stored within client device then client device and server device may immediately begin an updating process, or otherwise selectively establish a time in the future to begin such an updating process).

27. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Gu in view of Meyerson et al., (Pat. No. US 6,976,251 B2) (hereinafter 'Meyerson' - art made of record)

28. **As to claim 13**, Gu does not explicitly disclose the system wherein the electronic device further comprises an update agent, the update agent being capable of employing the reference software in conjunction with any retrieved update package to generate updated versions of the application software and also being capable of updating a plurality of application software employing reference software associated with each application software.

However, in an analogous art of Intelligent Update Agent, Meyerson discloses the system wherein the electronic device further comprises an update agent, the update agent being capable of employing the reference software in conjunction with any

Art Unit: 2192

retrieved update package to generate updated versions of the application software and also being capable of updating a plurality of application software employing reference software associated with each application software (e.g., Col. 4, Lines 58-63 – for the software update information to include a list of software updates available for software published by the publishers identified in the initial software update query).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Meyerson into the Gu's system to further provide the system wherein the electronic device further comprises an update agent, the update agent being capable of employing the reference software in conjunction with any retrieved update package to generate updated versions of the application software and also being capable of updating a plurality of application software employing reference software associated with each application software in Gu system.

The motivation is that it would further enhance the Gu's system by taking, advancing and/or incorporating Meyerson's system which offers significant advantages to provide a method of updating computer software that allows a user to determine in advance which software updates will be automatically installed and which will not be automatically installed depending on the criticality of a software update as once suggested by Meyerson (e.g., Col. 2, Lines 12-19).


Conclusion

29. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ben C. Wang whose telephone number is 571-270-1240. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.


TUAN DAM
SUPERVISORY PATENT EXAMINER

BCW 

August 14, 2007